

Exercícios - Ordenação Interna

[Iniciar tarefa](#)

Vencimento Sexta-feira por 23:59 **Pontos** 1 **Enviando** um upload de arquivo

1. Dada a sequência de números que compõe a sua matrícula acadêmica, ordene-a em ordem crescente segundo os seguintes algoritmos, apresentando a sequência obtida após cada passo do algoritmo. Para cada algoritmo mostre o custo computacional para o número de comparações e movimentações executadas.
 - a) *Bubble sort*
 - b) *Selection sort*
 - c) *Insertion sort*
 - d) *Shellsort*
 - e) *Mergesort*
 - f) *Quicksort*
2. Na implementação clássica do *Bubble sort*, um vetor é ordenado borbulhando o maior valor para o final do vetor a cada iteração. Uma ordenação por coqueteleira, idealizado por Donald Knuth, é uma modificação da ordenação do *Bubble sort*, na qual a direção do borbulhamento muda em cada iteração; em uma iteração, o menor elemento é borbulhado para o início; na seguinte, o maior é borbulhado para final do vetor; e assim por diante. Implemente este novo algoritmo e realize a análise de sua complexidade.
3. João diz ter desenvolvido um algoritmo que é capaz de ordenar qualquer conjunto de n números reais, fazendo apenas $O(n^{3/2})$ comparações. Você compraria este algoritmo? Justifique.
4. Escreva uma versão recursiva dos algoritmos de ordenação *Bubble sort*, *Selection sort* e *Insertion sort*.
5. O problema da ordenação parcial ocorre quando se deseja obter os k primeiros elementos de um arranjo contendo n elementos, em uma ordem ascendente ou descendente. Quando $k=1$, o problema se reduz a encontrar o mínimo (ou o máximo) de um conjunto de elementos. Quando $k = n$, caímos no problema clássico de ordenação. Altere os algoritmos de ordenação abaixo para que eles possam realizar de forma eficiente a ordenação parcial:
 - a) *Bubble sort*
 - b) *Selection sort*
 - c) *Insertion sort*

d) *Quicksort*